Research and Application of an Improved Support Vector Clustering Algorithm on Anomaly Detection

Sheng SUN[1, 2]
[1]School of Computer Science, Huazhong University of Science and Technology, Wuhan, China
[2]School of Computer Science, Huangshi Institute of Technology, Huangshi, China
Email: ss_hust@sina.com

YuanZhen WANG
School of Computer Science, Huazhong University of Science and Technology, Wuhan, China
Email: wangyz2005@163.com

***Abstract*—Anomaly detection approaches build models of normal data and detect deviations from the normal model in observed data. Anomaly detection applied to intrusion detection and computer security has been an active area of research. The major benefit of anomaly detection algorithms is their ability to potentially detect unforeseen attacks. In this paper, a novel weighted support vector clustering algorithm for anomaly detection is proposed. The weight to each input point is defined according to the position of samples in sphere space. The results of experiment demonstrate that the algorithm has excellent capability and applying it in intrusion detection system can be an effective way via using the data sets of KDD cup 99.**

***Index Terms*—clustering; support vector clustering; anomaly detection; outlier**

Ⅰ. INTRODUCTION

Over the past ten years, the number as well as the severity of network-based computer attacks has significantly increased. Attacks use weakness in the system to violate the policy, these weaknesses are called vulnerabilities. The sophistication of attacks increased with self replicating codes, such as viruses, then password cracking (where the cryptographic password is broken), and on to the more sophisticated threats shown. Against this rising sophistication in threats, we have easy availability of hacking tools: hackers no longer have to be experts in computer science or security; they could use available tools. For example, a tool such as nmap can be used to find all the open ports, a first stage in an attack. This combination of decreasing knowledge required of the attackers and the increasing sophistication of the attacks is giving rise to major security concerns.

Consider in business field, customers will submit information via the Internet only if they are confident that their private information, such as credit card numbers, is secure. Therefore, today's Web-based services must include solutions that provide security as a primary component in their design and deployment. Since traditional passive security such as firewalls and data encryption is insufficient to defend against attacks by crackers, Intrusion detection techniques which own active defense character have recently gained much attention. The goal of intrusion detection is to identity, preferably in real time, unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators. The intrusion detection problem is becoming a challenging task due to the proliferation of heterogeneous computer networks since the increased connectivity of computer systems gives greater access to outsiders and makes it easier for intruders to avoid identification.

As defined in [1], intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network. Various intrusion detection techniques are emerged in recent years, including neural network, genetic algorithm, fuzzy recognition, data mining, etc.

In data mining, clustering is a popular technique used to group similar data points or objects in groups or clusters. Traditionally clustering techniques are broadly divided in hierarchical and partitioning. Hierarchical clustering is further subdivided into agglomerative and divisive. While hierarchical algorithms build clusters gradually, partitioning

doi:10.4304/jsw.5.3.328-335

algorithms learn clusters directly. In doing so, they either try to discover clusters by iteratively relocating points between subsets, or try to identify clusters as areas highly populated with data. Partitioning algorithms are further categorized into probabilistic clustering (EM framework, algorithms SNOB, AUTOCLASS, MCLUST), k-medoids methods (algorithms PAM, CLARA, CLARANS, and its extension), and k-means methods. Such methods concentrate on how well points fit into their clusters and tend to build clusters of proper convex shapes.

Clustering methods have been dealt with the intrusion detection problem [2, 19, 20, 21, and 24], since they can help current intrusion detection systems in several respects. An important advantage of using clustering to detect network attacks is the ability to find new attacks not seen before. This implies that attack types with unknown pattern signatures can be detected.

Support vector clustering (SVC) originated from the concept of support vector machines. Kernel methods are the basis of the support vector clustering approach [6, 7], this method defines a nonlinear mapping between the input space and feature space through a kernel function, then find a hyper sphere with minimal radius that contains most of the mapped data points. The decision boundary in the input space is represented by a hyper sphere in the higher dimensional feature space. In [3], the thesis presents basic principal and advantages of SVC; however, the weakness of SVC algorithm is the high false alarm rate. This paper improves the traditional SVC algorithm and proposes a weighted support vector clustering algorithm, and applies the novel algorithm to anomaly intrusion detection. The main challenge of anomaly detection is to minimize false alarm rate. The improved SVC clustering algorithm in this paper is effective to reduce false positive rate with a desirable detection rate. The proposed approach is evaluated using the KDD'99 dataset, which were used for the third International Knowledge Discovery and Data Mining Tools Competition. Our experimental results show that experiments on real-time intrusion detection achieve higher attack detection rate with lower false positive rate.

The rest of this paper is organized as follows. The anomaly detection methods using clustering will be described in section Ⅱ. Section Ⅲ presents a weighted support vector clustering algorithm. In section Ⅳ, we introduce the new weighted SVC method to network intrusion detection and present the experimental performance evaluations upon KDD Cup'99 data set. Some concluding remarks are given in Section Ⅴ.

Ⅱ. ANOMALY DETECTION USING CLUSTERING

The existing intrusion detection methods fall in two major categories: signature detection (also known as misuse detection), and anomaly detection. Anomaly detection has been an important subject in intrusion detection research. Anomaly detection approaches build models of normal data and then attempt to detect deviations from the normal model in observed data [8, 9, 10, and 11]. These algorithms can detect known intrusions and new types of intrusions, because these new intrusions, by assumption, will deviate from normal network usage. This approach is complementary with respect to signature detection, where a number of attack descriptions are matched against the stream of audited events, looking for evidence that one of the modeled attacks is occurring [12, 13, 14].

Signature detection methods are well understood and widely applied. They are used in both host based systems, such as virus detectors, and in network based systems such as SNORT [25] and BRO [26]. These systems use a set of rules encoding knowledge gleaned from security experts to test files or network traffic for patterns known to occur in attacks. In spite of the fact that signature detection models have high degree of accuracy in detecting known attacks and their variations, their obvious drawback is the inability to detect attacks whose instances have not yet been observed. Misuse detection can only detect attacks that are well known and for which signatures have been written.

Anomaly detection is a harder problem than signature detection because while signatures of attacks can be very precise, what is considered normal is more abstract and ambiguous. In anomaly detection problem, given a set of normal data to train from, and given a new piece of test data, the goal of the intrusion detection algorithm is to determine whether the test data belong to “normal” or to an anomalous behavior. Though anomaly detection can detect novel attacks, it has the drawback of not capable of discerning intent; it can only signal that some event is unusual, but not necessarily hostile, thus generating false alarms.

There are two major categories of anomaly detection techniques, namely supervised and unsupervised. In supervised anomaly detection, given a set of normal data to train on, and given a new set of test data, the goal is to determine whether the test data is ‘normal’ or anomalous. Recently, there have been several efforts in designing supervised network-based anomaly detection algorithms, such as ADAM [28], PHAD [31], NIDES [27], and other techniques that use neural networks, information theoretic measures, network activity models, etc. Unlike supervised anomaly detection where the models are built only according to the normal behavior on the network, unsupervised anomaly detection attempts to detect anomalous behavior without using any knowledge about the training data. Unsupervised anomaly detection approaches are based on statistical approaches [15], clustering [33], outlier detection schemes [22, 34], state machines [32], etc.

Most data sets contain one or a few unusual observations that do not seem to belong to the pattern of variability produced by other observations. When an observation is different from the majority of the data or is sufficiently unlikely under the assumed probability model of the data, it is considered an outlier. Outliers are likely indicating some special condition information. In unsupervised anomaly detection approaches, they detect attacks by determining unusual activities from data under two assumptions. The first assumption is that the number of normal instances vastly outnumbers the number of anomalies. The second assumption is that the anomalies themselves are qualitatively different from the normal instances. The basic idea is that since the anomalies are both different from normal and are rare, they will appear as outliers in the data which can be detected. If the two assumptions can not be attained, then UAD algorithms will be poor in detecting novel intrusions. For example, an unsupervised algorithm will have a difficulty detecting in a syn-flood DoS attack. The reason is that often under such an attack there are so many instances of the intrusion that it occurs in a similar number to normal instances. Thus, UAD algorithms may not label these instances as an attack because the region of the feature space where they occur may be as dense as the normal regions of the feature space.

Unlike standard supervised support vector machines that require labeled training data to create their classification rule, the SVC algorithm belongs to unsupervised learning algorithm. It does not require training data to be labeled to determine a decision surface. The idea of anomaly detection method using SVC algorithm is: First map the data points via kernel function to a new high-dimensional feature space, then find outliers according to the position of points in feature space, thus we can detect anomaly intrusions [16].

Whereas the supervised SVM algorithm tries to maximally separate two classes of data in feature space by a hyper plane, the unsupervised algorithm attempts to separate the entire set of training data from the origin, to find a small region where most of the data lies and label data points in this region as one cluster. Points in other regions are labeled as another cluster.

Intrusion detection systems (IDSs) are specific tools that are looking for malicious behavior, and when detecting it, they can take immediate appropriate action to prevent the attack. The action may be to send an alert, redirect the attack, or prevent the malicious behavior, and if the threat is a high risk, the IDS will alert the appropriate people like system administrators. IDSs can be used to detect a wide range of those events like: password cracking, protocol attacks, buffer overflows, illegal data manipulation, unauthorized file access, and many others. There exist two different sorts of network intrusion detection systems: signature-based and anomaly-based; both types are affected by a high false positive rate.

Anomaly detection systems (ABSs) first capture the network traffic and constructs dataset by pre-processing. After that, the statistical patterns are built over the dataset using clustering algorithm, and then flag any behavior that significantly deviates from the patterns as an attack. With the built patterns, we can find outliers related to each pattern. This is achieved by implementing a distance function and setting a threshold value: when the distance between the input sample and the pattern exceeds the threshold, then the sample is considered as outliers and the system will raise alerts.

After capturing network traffic and performing normalization on the data, we can cluster very large datasets with high dimensionality produced by network. Data transformation such as normalization may improve the accuracy and efficiency of mining algorithms involving neural networks, nearest neighbor and clustering classifiers. Such methods provide better results if the data to be analyzed have been normalized, that is, scaled to specific ranges such as [0.0, 1.0]. For distanced-based methods, normalization helps prevent attributes with initially large ranges from outweighing attributes with initially smaller ranges [17].

There are many methods for data normalization includes min-max normalization, z-score normalization and normalization by decimal scaling. We use max-min normalization method in this paper, which is shown in Eq. (1). The method performs a linear transformation on the original data. Suppose that $\min_a$ and $\max_a$ are the minimum and the maximum values for attribute A. Min-max normalization maps a value v of A to v' in the range [new-$\min_a$, new-

$\max_a$].

$$v' = \frac{v-\min_a}{\max_a-\min_a}(new\text{-}\max_a - new\text{-}\min_a) \qquad (1)$$

Normalization can change the original data and it is necessary to save the normalization parameters (the minimum and the maximum values) so that future data can be normalized in the same manner.

III. A WEIGHTED SUPPORT VECTOR CLUSTERING ALGORITHM

The mathematical formulation of weighted SVC algorithm is as follows. Assume given a data set $\{x_i\} \subseteq \chi$ of N points, with $\chi \subseteq R^d$, the input data space. **A** nonlinear mapping function φ is used to map χ into a high-dimensional feature space. The objective is to search for the smallest enclosing sphere of radius R expressed in the following optimization problem:

$$\min_{R,a,\xi_i} R^2 + C\sum_i s_i\xi_i$$

$$\text{Subject to } \|\phi(x_i)-a\|^2 \le R^2 + \xi_i \qquad (2)$$
$$\xi_i \ge 0, \forall i$$

Where R is the radius and the center of the enclosing sphere is the point **a**; ξ_i is a slack variable; allowing for soft boundaries; C is a constant that determines the tradeoff between the volume of the hyper sphere and the number of the outliers; s_i is the weight of a certain input sample point.

To solve this problem we introduce the Lagrangian:

$$L = R^2 - \sum_i (R^2 + \xi_i - \|\phi(x_i) - a\|^2)\beta_i - \sum_i \xi_i\mu_i + C\sum_i s_i\xi_i \qquad (3)$$

Where $\beta_i \ge 0, \mu_i \ge 0$ are Lagrange multipliers, and $c\sum_i s_i\xi_i$ is a penalty term.

The Wolf Dual [4, 5] optimization problem is

$$\max_{\beta_i} W = \sum_i \phi(x_i)^2\beta_i - \sum_{i.j}\beta_i\beta_j\phi(x_i)\cdot\phi(x_j)$$

$$s.t. \ \sum_i \beta_i = 1 \text{ and } 0 \le \beta_i \le s_iC \qquad (4)$$

And the Karush–Kuhn–Tucker conditions [6]:

$$\xi_i\mu_i = 0$$

$$(R^2 + \xi_i - \|\phi(x_i) - a\|^2)\beta_i = 0 \qquad (5)$$

In Eq. (4), the dot product $\phi(x_i)\cdot\phi(x_j)$ represents the Gaussian kernel transformation by $K(x_i,x_j) = e^{-q\|x_i-x_j\|^2}$ with width parameter q. The Lagrangian W now can be written as:

$$W = \sum_i K(x_i,x_i)\beta_i - \sum_{i,j}\beta_i\beta_j K(x_i,x_j) \qquad (6)$$

The distance between a point in the feature space and spherical center is:

$$R^2(x) = \|\phi(x) - a\|^2$$
$$= K(x,x) - 2\sum_i \beta_i K(x_i,x) + \sum_{i,j}\beta_i\beta_j K(x_i,x_j) \qquad (7)$$

Data points lying on the surface of the sphere in the feature space are called support vectors (SVs) when $0 < \beta_i < s_iC$. SVs can be used to describe the hyper sphere in the feature space. A point x_i is outside the sphere if the corresponding slack variable $\xi_i > 0$. From the KKT condition we know that $\beta_i = s_iC$, thus, we refer to those points as bounded support vectors (BSVs). Those points do not exist when C≥1. Similarly, a point x_i located inside the sphere when $\beta_i = 0$.

The radius **R** of the sphere can be obtained by

R= { $R(x_i)$ | x_i is a support vector} (8)

A point **x** is an outlier if R(x) > *R*.

We can extend the single cluster case to a multiple cluster problem by using the method given in [3]. We apply a weight to each input point of SVC and reformulate SVC into weighted SVC such that different input points can make different contributions to the learning of cluster boundaries in high dimensional feature space. Smaller values of s_i make the corresponding point x_i less important in the training. The weight s_i is defined by not only the relation between a sample and its cluster center but also by the affinity among samples. The affinity among samples is defined using a sphere with minimum volume while containing the maximum of the samples. We use two different ways to compute weight of samples inside sphere and out sphere respectively. The affinity-based weight computation formula is

$$s_i = \begin{cases} 0.6*(\dfrac{1-d(x_i)/R}{1+d(x_i)/R}) + 0.4, d(x_i) \le R \\ 0.4*(\dfrac{1}{1+(d(x_i)-R)}), d(x_i) > R \end{cases} \qquad (9)$$

Where R is the spherical radius in feature space; $d(x_i)$ is the distance between x_i and the spherical center **a** in feature space, from (7), one can compute the distance between x and spherical center **a** in feature space as

$$d(x_i) = \left\| \phi(x_i) - a \right\|$$
$$= (K(x,x) - 2\sum_i \beta_i K(x_i, x) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j))^{1/2}$$
$$i = 1, \cdots, N$$
$$(10)$$

Outliers are located outside the hyper sphere in the feature space; the weight of each outlier is lower than 0.4, and the lower the weight of sample, the more possibility that the sample is outlier.

Ⅳ. EXPERIMENTS

We tested our new weighted support vector clustering methods using the 1999 KDD Cup network intrusion dataset [18]. It originated from the 1998 DARPA Intrusion Detection Evaluation Program managed by MIT Lincoln Labs. It contained a wide variety of intrusions simulated in a military network environment. The DARPA'98 data contains two types: training data and test data. The training set consisted of approximately 4,900,000 data instances, each of which is a vector of extracted feature values from a connection record obtained from the raw network data gathered during the simulated intrusions. The testing set consists of two weeks of traffic with around 300,000 connections. A connection is a sequence of TCP packets to and from some IP addresses. The TCP packers were assembled into connection records using the Bro program modified for use with MADAM/ID. Each connection was labeled as either normal or as exactly one specific kind of attack. All labels are assumed to be correct. The training set was used to set parameters values for the algorithm and the second, the test set, was used for evaluation.

The data contains four main categories of attacks: DoS, R2L, U2R and PROBING. DoS (Denial of Service), for example, ping-of-death, teardrop, smurf, SYN flood, etc.; R2L, unauthorized access from a remote machine, for example, guessing password; U2R, unauthorized access to local super user privileges by a local unprivileged user, for example, various buffer overflow attacks; PROBING, surveillance and probing, for example, port-scan, ping-sweep, etc.

In a Denial-of-Service (DoS) attack, the attacker attempts to use up all the victim system's resources like memory or bandwidth. When the attack is successful, legitimate users can no longer access the resources and the services offered by the server will be shut down. According to the 2002 CSI/FBI survey, 40% of all attacks are DoS attacks. An attack can be directed at an operating system or at the network. The attacker may send specially crafted packets that crash remote software/services running on the victim server. It will be successful if the network is unable to distinguish between legitimate traffic and malicious or bogus traffic.

While the detection system was able to detect most types of attacks well, it was not able to detect DoS attack well because this type of attack using the feature space was in the same region as normal data.

The experiment procedures consist of three steps: First, preprocess the captured network data packets and constructs dataset; then divide original data into training data and test data; lastly, use our new clustering method to label network connection records, and give the detection threshold, according to the threshold computed the detection rate and false positive rate.

A. Feature construction

Each record collected in the experiments contains 41 features describing the connection, with most of them taking on continuous values. It is labeled either normal or one of the attack types. The 41 features can be divided into three groups: the first group is the basic features of individual TCP connections, for example, duration, protocol type, number of bytes transferred, the flag indicating the normal or error status. The second group is the content features within a connection suggested by domain knowledge, such as the number of file creation operations, number of failed login attempts. And the third group is the traffic features computed using a two-second time window. Many of these features are redundant, so we only select 18 features which can reflect the behavior of users as research objects. They are duration, protocol_type, service, src_bytes, and dst_bytes, etc. A description of the 18 selected features is listed in Table 1.

Although we have use all three groups of features for our experiments, it is well known that constructed features from the data content of the connections are more important when detecting R2L and U2R attack types, while "time-based' and "connection-based" features were more important for detection DoS and probing attack types[23].

Table 1. Feature description of network connection records

Feature Name	**Description**
duration	of seconds of the connection
protocol type	Type of the protocol

service	Network service on the destination
flag	Normal or error status of the connection
src bytes	of data bytes from source to destination
dst bytes	of data bytes from destination to source
land	1: connection is from or to the same host or port; 0: otherwise
Wrong fragment	of “wrong” fragments
urgent	of urgent packets
count	of connections to the same host
srv_count	of connections to the same service
serror_rate	of connections with SYN errors to the same host
srv_serror_rate	of connections with SYN errors to the same service
rerror_rate	of connections with REJ errors to the same host
srv_rerror_rate	of connections with REJ errors to the same service
same_srv_rate	of connections to the same service
diff_srv_rate	of connections to different services
srv_diff_host_rate	of connections to different hosts

As mentioned in section Ⅱ, unsupervised anomaly detection algorithms are sensitive to the ratio of intrusions in the data set. If the number of intrusions is too high, each intrusion will not show up as anomalous. In order to make the data set more realistic we filtered many of the attacks so that the resulting data set consisted of 1 to 2% attack and 98 to 99% normal instances.

To meet the needs of two hypothesizes in the detection algorithm, we sample five groups of data from the whole detection dataset for experimental data, each group contains 158000 records, including 154900 normal records and 3100 anomalies, the percentage of normal records in each group is more than 98%, the amount of normal data is much more than anomalies.

B. Experimental Results

The accuracy of an intrusion detection system is measured by its detection rate and its false alarm rate. The detection rate (DR) is defined as the number of intrusion instances detected by the system divided by the total number of intrusion instances present in the test set. The detection rate of our detection approach over the KDD’99 testing data is computed by using Equation (11).

$$DR = \frac{\sum_{service=1}^{n} \text{no. of detected attacks}}{\sum_{service=1}^{n} \text{no. of total attacks}} \tag{11}$$

The false alarm rate (FPR) is defined as the total number of normal instances that were (incorrectly) classified as intrusions divided by the total number of normal instances. We calculate the false positive rate of the detection approach using Equation (12).

$$FPR = \frac{\sum_{service=1}^{n} \text{no. of false alarms}}{\sum_{service=1}^{n} \text{no. of total normal records}} \tag{12}$$

These are good indicators of performance, since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications it makes in the process. A good method should provide a high Detection Rate together with a low False Alarm Rate. These values are calculated over the labeled data to measure performance.

The experimental results are presented in Table 2.

Table 2. Average detection rates from the weighted SVC

Sample Set	Detection (%)	False Alarm Rate (%)
First Group	98.81	0.91
Second Group	99.06	0.75
Third Group	99.37	0.48
Fourth Group	99.14	0.60
Fifth Group	99.25	0.72
The Whole	99.12	0.69

As shown in Table 2, our proposed weighted support vector clustering scheme works effectively in identifying the attacks. Most attacks can be distinguished from the normal activities and its detection rate is as high as 99.12%. At the same time, the false alarm rate as low as 0.69%.

Plotting the detection rate as a function of the false alarm rate, we end up with what is called a ROC (Receiver Operating Characteristic) curve. ROC curves are plotted depicting the relationship between false positive and detection rates for one fixed training/test set combination. ROC curves are a way of visualizing the trade-off between detection and false positive rates. The nearer the ROC curve of a scheme is to the upper-left corner, the better the performance of the scheme is.

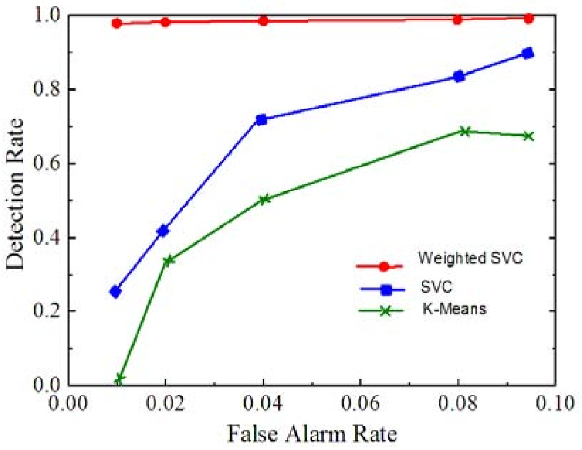

Figure 1. ROC curves of three anomaly detection methods

Compared to other unsupervised anomaly based systems, such as k-means and SVC, our proposed weighted SVC approach performs best, as shown from the ROC curves in Fig. 1. It is the only method that works well at low false alarm rate. Its stability and reliability is apparently better than the other detection methods.

Ⅴ. CONCLUSIONS

The paper proposed a novel weighted support vector clustering algorithm, and it can cluster large data set and high-dimensional data effectively. As more and more organizations become vulnerable to a wide variety of cyber threats, it is important to have an efficient algorithm that is capable of distinguishing between the attacks and normal connections. We have also introduced the proposed SVC method to network intrusion detection. The experiments with KDD Cup 1999 data demonstrate that our proposed method has efficient performance in network intrusion detection.

ACKNOWLEDGMENT

It is a pleasure to acknowledge the contributes of Hong Zhu and Wan-Shan Zhang, both of whom participated in a number of fruitful discussions. The authors thank the Database and Multimedia Technology Research Institute for their support of this research through grant 0487032 to Huazhong University of Science and Technology.

REFERENCES

[1] R. Bace and P. Mell, “Intrusion Detection Systems”, *NIST Special Publications SP 800-31*, November, 2001.

[2] Y. Guan, A. A. Ghorbani, and N. Belacel, "Y-MEANS: a clustering method for intrusion detection," in *Canadian Conference on Electrical and Computer Engineering*, pp. 1083-1086, 2003.

[3] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," in *J. Machine Learning Research*, vol. 2, pp. 125-137, 2001.

[4] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.

[5] V. N. Vapnik, The Nature of Statistical Learning Theory. *New York: Springer-Verlag*, 1995.

[6] Nello Cristianini, John Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. *Cambrige University Press*, 2000.

[7] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on COLT,* pages 144-1 52. ACM Press,1998.

[8] D.E. Denning. An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, 13(2):222-232, February 1987.

[9] A.K. Ghosh, J. Wanken, and F. Charron. Detecting Anomalous and Unknown Intrusions Against Programs. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC'98)*, pages 259-267, Scottsdale, AZ, December 1998.

[10] C. Ko, M. Ruschitzka, and K. Levitt. Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specication-based Approach. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 175-187, May 1997.

[11] T. Lane and C.E. Brodley. Temporal sequence learning and data reduction for anomaly detection. In *Proceedings of the 5th ACM conference on Computer and communications security*, pages 150-158. ACM Press, 1998.

[12] K.Ilgun, R.A.Kemmerer, and P.A.Porras. State Transition Analysis: A Rule-Based Intrusion Detection System. *IEEE Transactions on Software Engineering*, 21(3):181-199, March 1995.

[13] U. Lindqvist and P.A. Porras. Detecting Computer and Network Misuse with the Production-Based Expert System Toolset(P-BEST). In *IEEE Symposium on Security and Privacy*, pages 146-161, Oakland, California, May 1999.

[14] V. Paxson. Bro: A System for Detecting Network Intruders in Real-Time. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, January 1998.

[15] Stuart Staniford, James A. Hoagland, and Joseph M. McAlerney. Practical automated detection of stealthy portscans. *Journal of Computer Security*, 10:105–136, 2002.

[16] Eskin E and Arnold A.Geometric Framework for Unsupervised Anomaly Detection: Detection Intrusions in Unlabeled Data. *Data Mining for Security Applications (DMSA-2002)*, Kluwer, 2002.

[17] Jiawei Han, Micheline Kamper. Data Mining: Concepts and techniques [M]. *Morgan Kaufmann Publisher*, 2000, 9-10.

[18]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[19] Shi Zhong, Taghi M. Khoshgoftaar, and Naeem Seliya. Evaluating clustering techniques for network intrusion detection. In *10th ISSAT Int. Conf. on Reliability and Quality Design*, pages 173-177, Las Vegas, Nevada, USA, August 2004.

[20] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proc. 17th Int. Conf. Machine Learning*, pages 255-262, San Francisco, CA, 2000.

[21] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *ACM Workshop on Data Mining Applied to Security*, Philadelphia, PA, November 2001.

[22] Markus Breunig, Hans-Peter Kriegel, Raymond T. Ng, and J¨org Sander. Lof: Identifying densitybased local outliers. In *Proceedings of the ACM SIGMOD Conference*, Dallas, TX, 2000.

[23] W. Lee, S. J. Stolfo, Data Mining Approaches for Intrusion Detection, *Proceedings of the 1998 USENIX Security Symposium*, 1998.

[24] E. Leon, 0. Nasraoui, and J. Gomez, "Anomaly detection based on unsupervised niche clustering with application to network intrusion detection," in *Proceedings of IEEE Conference on Evolutionary Computation (CEC)*, pp. 502-508, 2004.

[25] M. Roesch. Snort-lightweight intrusion detection for networks. In *USENIX LISA*, 1999.

[26] V. Paxon. Bro: A system for detecting network intruders in real-time. In *Proc. 7th USENIX Security Symp.*, 1998.

[27] D. Anderson, T. Lunt, H. Javitz, A. Tamaru, and A. Valdes. Detecting unusual program behavior using the statistical component of the next-generation intrusion detection expert system (NIDES). Technical Report SRI-CSL-95-06, SRI, 1995.

[28] D. Barbara, N. Wu, and S. Jajodia. Detecting novel network intrusions using bayes estimators. In *Proc. SIAM Intl. Conf. Data Mining*, 2001.

[29] H. S. Javitz and A. Valdes. The SRI Statistical Anomaly Detector. *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy*, May 1991.

[30] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. P. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman. Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation. *Proceedings DARPA Information Survivability Conference and Exposition (DISCEX) 2000*, Vol. 2, pp. 12-26, IEEE Computer Society Press, CA, 2000.

[31] Matthew V. Mahoney and Philip K. Chan. PHAD: Packet header anomaly detection for identifying hostile network traffic. Technical report, Florida Tech., 2001.

[32] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specification based anomaly detection: A new approach for detecting network intrusions. In *ACM Conference on Computer and Communications Security*, 2002.

[33] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Data Mining for Security Applications*, 2002.

[34] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD Conference*, pages 427–438, Dallas, TX, 2000.

Sheng Sun was born in 1978. He is currently working toward the Ph.D. degree at the Computer Science Department, Huazhong University of Science and Technology (HUST), Wuhan, China. His present research interests include databases and data mining.

YuanZhen Wang is currently professor and PHD supervisor in the college of Computer Science and Technology at HUST. Her research interests include distributed multimedia database management system, object oriented databases, data mining, data warehouse, mobile database.